

A Sparse and Low-Rank Optimization Framework for Index Coding via Riemannian Optimization

Yuanming Shi^α and Bamdev Mishra^β

^αSchool of Information Science and Technology, ShanghaiTech University, Shanghai, China

E-mail: shiym@shanghaitech.edu.cn

^βAmazon Development Centre India, Bangalore, Karnataka 560055, India

E-mail: bamdevm@amazon.com

Abstract—Side information provides a pivotal role for message delivery in many communication scenarios to accommodate increasingly large data sets, e.g., caching networks. Although index coding provides a fundamental modeling framework to exploit the benefits of side information, the index coding problem itself still remains open and only a few instances have been solved. In this paper, we propose a novel sparse and low-rank optimization modeling framework for the index coding problem to characterize the tradeoff between the amount of side information and the achievable data rate. Specifically, sparsity of the model measures the amount of side information, while low-rankness represents the achievable data rate. The resulting sparse and low-rank optimization problem has non-convex sparsity inducing objective and non-convex rank constraint. To address the coupled challenges in objective and constraint, we propose a novel Riemannian optimization framework by exploiting the quotient manifold geometry of fixed-rank matrices, accompanied by a smooth sparsity inducing surrogate. Simulation results demonstrate the appealing sparsity and low-rankness tradeoff in the proposed model, thereby revealing the tradeoff between the amount of side information and the achievable data rate in the index coding problem.

I. INTRODUCTION

With the dramatic increase of smart mobile devices, as well as diversified services and applications, we are in the era of data deluge [1]. Meanwhile, with the emerging applications empowered by the Internet of Things (IoT) and Tactile Internet, massive devices will need to get connected, which calls for ultra-low latency, high availability, reliability and security communications [2]. However, with the low latency and high data rate requirements, the communication systems are placed under tremendous pressure to accommodate increasingly large data sets and to efficiently deliver the content. To resolve the big data challenge in communication networks, side information plays a pivotal role for both the wired and wireless communication links to deliver messages to users [3], [4]. That is, users can access to the messages as the side information that requested by other users. For instance, this scenario arises in the cache enabled fog radio access networks (Fog-RAN) [5]. In this network architecture, the content can be stored in the caches or other storage elements, e.g., the fog data center, radio access points and the mobile devices. The cached content may be requested by other users or in the further, thereby providing side information for message delivery in wired and wireless communications [3], [6].

Index coding provides a powerful framework to model the communication scenarios with side information [4]. Although it has been shown that the index coding problem is related to many challenging problems (e.g., distributed storage, topological interference management [7] and network coding [8]), the index coding problem itself remains open. Most of works on index coding focus on how to exploit the fixed side information, thereby designing efficient message delivering strategies, e.g., the interference alinement approach [4]. In particular, in caching networks [3], the side information (i.e., message placement) can be designed, followed by the message delivery. However, the amount of the side information in caching networks is limited by the storage capacity in caching networks.

In this paper, we put forth a different viewpoint on the index coding problem by investigating the fundamental tradeoff between the amount of the side information and the achievable data rate. That is, the higher data rate comes from at the price of high storage size, yielding more side information. To achieve this goal, we propose a novel sparse and low-rank optimization framework to minimize the amount of side information to meet a data rate requirement. Specifically, the sparsity of this model represents the amount of side information, while the low-rankness of this model represents the number of channel uses, i.e., blocklength, which equals the inverse of the achievable data rate. Although the sparse and low-rank models have recently been well-studied in signal processing and machine learning [9], the presented model for index coding is novel and can help reveal the fundamental tradeoff between the amount of side information and the achievable data rate.

Unfortunately, the resulting sparse and low-rank optimization problem raises a unique challenge due to a non-convex objective function (ℓ_0) and non-convex constraint (rank). Although the convex relaxation approach based on convex surrogates – ℓ_1 -norm and nuclear norm – can provide polynomial time complexity algorithms [9], this approach is inapplicable in our problem as it always return the identity matrix. Another approach is based on alternating minimization by factorizing a fixed-rank matrix [10], accompanied with ℓ_1 -norm relaxation. However, this approach fails to yield good performance by inducing a less sparse solution and is computationally expensive using the off-the-shelf parser/solver CVX [11].

To address the limitations of the above methods, we propose

a Riemannian optimization algorithm [12] to solve the resulting sparse and low-rank optimization problem. In particular, by exploiting the quotient manifold geometry of fixed-rank matrices [13], the Riemannian optimization algorithm was proposed to solve the low-rank matrix completion problem for topological interference management [7]. However, this algorithm can not be applied in our problem due to the additional affine constraint preserving the desired signals and the non-convex sparsity inducing objective. We thus propose a smooth sparsity inducing surrogate and regularize the affine constraint as a smooth least-squares term. The second-order trust-region method [12] is further applied to the resulting optimization problem with smooth objective over fixed-rank manifold constraint. The proposed algorithm, which is implemented in the manifold optimization toolbox Manopt [14], outperforms the alternating minimization algorithm in terms of implementation complexity and performance. Simulation results demonstrate the appealing tradeoff between the sparsity and low-rankness of the model, thereby revealing the tradeoff between the amount of side information and the achievable data rate.

II. PROBLEM STATEMENT

We consider the communication networks (e.g., caching network [3]) with side information to help message delivery. To investigate the tradeoff between the amount of side information and the achievable data rate, we introduce an index coding modeling framework for communications with side information [4]. Specifically, we consider a multiple unicast index coding problem consists of a set of K independent messages W_1, W_2, \dots, W_K , and a set of K destination nodes. The i -th destination desires message W_i with side information index as \mathcal{V}_i and $i \notin \mathcal{V}_i$.

Let \mathcal{S} be the choice of a finite alphabet. The coding function f for all the messages is given by $f(W_1, W_2, \dots, W_K) = \mathbf{z}$, where $\mathbf{z} \in \mathcal{S}^N$ is the sequence of symbols transmitted over N channel uses. Here, each message W_i is a random variable uniformly distributed over the set $W_i \in \{1, 2, \dots, |\mathcal{S}|^{NR_i}\}$ with $|\mathcal{S}|^{NR_i}$ as an integer. At destination i , the decoding function g_i for the desired message W_i is given by $g_i(\mathbf{z}, \mathcal{V}_i) = \hat{W}_i$. The probability of decoding error is given by $p_e = 1 - \Pr\{\hat{W}_i = W_i, \forall i\}$.

Define the above coding scheme as $(\mathcal{S}, N, (R_1, \dots, R_K))$. If for every $\epsilon, \delta > 0$, for some \mathcal{S} and N , there exists a coding scheme $(\mathcal{S}, N, (\bar{R}_1, \bar{R}_2, \dots, \bar{R}_K))$, such that $\bar{R}_i \geq R_i - \delta, \forall i$, and the error probability $P_e \leq \epsilon$, then the rate tuple $(R_1, R_2, \dots, R_K) \in \mathbb{R}_+^K$ is said to be achievable. Note that the index coding capacity does not dependent on the field specification [4]. In this paper, the achievable scheme is restricted to the real field \mathbb{R} for linear coding schemes design to construct index codes over real field.

A. Scalar Linear Index Coding Scheme

Consider a scalar linear index coding scheme, which sends one symbol for each message over N channel uses. Let $\mathbf{v}_i \in \mathbb{R}^N$ and $\mathbf{u}_i \in \mathbb{R}^N$ be the precoding vector and the decoding vector, respectively. The transmitted symbol sequence

$\mathbf{z} \in \mathbb{R}^{N \times 1}$ over N channel uses in a linear coding scheme is given by $\mathbf{z} = \sum_{i=1}^K \mathbf{v}_i s_i$, where s_i is one symbol from \mathbb{R} representing W_i . The decoding operation for message W_k at destination k is given by

$$\hat{s}_k = (\mathbf{u}_k^T \mathbf{v}_k)^{-1} \mathbf{u}_k^T \left(\mathbf{z} - \sum_{i \in \mathcal{V}_k} \mathbf{v}_i s_i \right). \quad (1)$$

The above decoding operation is achieved by the following interference alignment condition [4], [7]:

$$\mathbf{u}_k^T \mathbf{v}_k \neq 0, \forall k = 1, \dots, K \quad (2)$$

$$\mathbf{u}_k^T \mathbf{v}_i = 0, \forall i \neq k, i \notin \mathcal{V}_k. \quad (3)$$

If the above interference alignment conditions (2) and (3) are satisfied over N channel uses, the following data rate vector $\mathcal{R} = (\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N})$, can be achieved [4], [7]. Therefore, the achievable sum data rate is given by K/N .

B. Storage Size and Data Rate Tradeoff

We shall give an example of caching network to illustrate the amount of side information and achievable data rate. Specifically, we assume that the file library is a set of K messages $\{W_1, \dots, W_K\}$, where each has entropy F bits. Given the side information \mathcal{V}_i 's after content placement, the amount of side information is $\sum_{i=1}^K |\mathcal{V}_i| F$ bits, which measures the total storage size. In this case, we characterize the tradeoff between the following two important metrics:

- The amount of side information: $s := \sum_{i=1}^K |\mathcal{V}_i|$ (normalized by F);
- The achievable data rate: $r := 1/N$ (normalized by K).

In general, the more side information is available, the higher data rate can be achieved.

The index coding problem has shown to be related to the network coding problem [8], topological interference management problem [7], caching problem [3], as well as distributed storage problem. This paper thus can provide principles for all these important design problems by characterizing the tradeoff between the amount of the side information and the achievable data rate.

III. A SPARSE AND LOW-RANK OPTIMIZATION FRAMEWORK FOR INDEX CODING

In this section, we propose a unified sparse and low-rank modeling framework to investigate the tradeoffs between the amount of side information $\sum_i |\mathcal{V}_i|$ and the achievable data rate $1/N$ in the index coding problem. This is achieved by rewriting the interference alignment conditions (2) and (3) into a sparse minimization problem with a fixed-rank constraint and an affine constraint.

A. Sparse and Low-Rank Modeling Framework

Let $X_{ij} = \mathbf{u}_i^T \mathbf{v}_j, \forall i, j = 1, \dots, K$. Define the $K \times K$ matrix $\mathbf{X} = [X_{ij}]$, we have the rank of matrix \mathbf{X} as $\text{rank}(\mathbf{X}) = N$. The achievable data rate (normalized by K) is given by

$$r = 1/\text{rank}(\mathbf{X}). \quad (4)$$

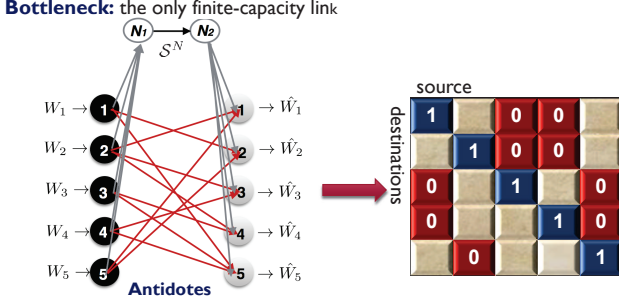


Fig. 1. (a) The index coding problem with only one finite capacity link and all the other links having infinite capacity. The side information is given by $\mathcal{V}_1 = \{2, 5\}$, $\mathcal{V}_2 = \{1, 5\}$, $\mathcal{V}_3 = \{2, 4\}$, $\mathcal{V}_4 = \{2, 3\}$, $\mathcal{V}_5 = \{1, 3, 4\}$. (b) The associated incomplete matrix representing the interference alignment conditions (2) and (3).

Additionally, the sparsity of the matrix \mathbf{X} is given by $\|\mathbf{X}\|_0 = \sum_{i=1}^K |\mathcal{V}_i| + K$. Finally, the amount of side information (normalized by F) is given by

$$s = (\|\mathbf{X}\|_0 - K). \quad (5)$$

An example of the sparsity and low-rankness of the matrix \mathbf{X} for the index coding problem is shown in Fig. 1. In this case, the amount of side information is given by $s = \sum_i |\mathcal{V}_i| = 11$, which equals $(\|\mathbf{X}\|_0 - K) = 11$ by assuming that the unknown entries in the associated incomplete matrix are non-zero.

From (4) and (5), we can see that, to characterize the tradeoff between the amount of side information (i.e., storage size) and the achievable data rate, it is equivalent to characterize the tradeoff between the sparsity and low-rankness of the modeling matrix \mathbf{X} . Specifically, we propose to solve the following sparse and low-rank optimization problem:

$$\begin{aligned} \mathcal{P} : \text{minimize} \quad & \|\mathbf{X}\|_0 \\ \text{subject to} \quad & X_{ii} = 1, \forall i = 1, \dots, K \\ & \text{rank}(\mathbf{X}) = r, \end{aligned} \quad (6)$$

where r is a fixed rank value of matrix \mathbf{X} . By solving a sequence of the optimization problem \mathcal{P} via varying r from 1 to K , we can reveal the tradeoff between the sparsity and low-rankness of matrix \mathbf{X} .

B. Problem Analysis

The widely used ℓ_1 -norm and nuclear-norm relaxation method provides a computationally tractable algorithm for the sparse and low-rank optimization as follows [9]:

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{X}\|_1 + \lambda \|\mathbf{X}\|_* \\ \text{subject to} \quad & X_{ii} = 1, \forall i = 1, \dots, K, \end{aligned} \quad (7)$$

where $\lambda \geq 0$ is a regularized parameter, $\|\mathbf{X}\|_1 := \sum_{ij} |X_{ij}|$, and $\|\mathbf{X}\|_*$ is the nuclear norm of \mathbf{X} , i.e., it is defined as the summation of the singular values of \mathbf{X} . $\|\mathbf{X}\|_1$ and $\|\mathbf{X}\|_*$ are popular convex surrogates of $\|\mathbf{X}\|_0$ and the rank constraint, respectively. Unfortunately, since $\|\mathbf{X}\|_* \geq |\text{Tr}(\mathbf{X})|$ [7] and $\|\mathbf{X}\|_1 \geq K$, the problem (7) always returns $\mathbf{X} = \mathbf{I}_K$ as solution, which clearly is not low rank.

Another approach is based on alternating minimization by factorizing the rank- r matrix \mathbf{X} as \mathbf{UV}^T , where $\mathbf{U} \in \mathbb{R}^{K \times r}$ and $\mathbf{V} \in \mathbb{R}^{K \times r}$ are full column rank matrices. Consequently, problem \mathcal{P} is further relaxed as follows:

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{UV}^T\|_1 \\ \text{subject to} \quad & [\mathbf{UV}^T]_{ii} = 1, \forall i = 1, \dots, K, \end{aligned} \quad (8)$$

where $[\cdot]_{ij}$ denotes the (i, j) -entry of a matrix. The alternating minimization algorithm for problem (8) consists of alternatively solving for \mathbf{U} and \mathbf{V} while fixing the other factor. However, the alternating minimization algorithm fails to exploit the second-order information to improve the performance, i.e., enhance sparsity in matrix \mathbf{X} .

In this paper, in order to enhance sparsity via exploiting the second-order information, we propose a Riemannian optimization algorithm to approximately solve problem \mathcal{P} .

IV. RIEMANNIAN OPTIMIZATION ALGORITHM

In this section, we propose a Riemannian optimization algorithm to solve problem \mathcal{P} . Specifically, the ℓ_0 -norm is relaxed to the ℓ_1 -norm, resulting in the optimization problem:

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{X}\|_1 \\ \text{subject to} \quad & X_{ii} = 1, \forall i = 1, \dots, K \\ & \text{rank}(\mathbf{X}) = r. \end{aligned} \quad (9)$$

However, the intersection of rank constraint and the affine constraint is challenging to characterize. We, therefore, propose to solve (9) in two steps. In the first step, we find a good sparsity pattern by considering a regularized version of (9). In the second step, we refine the estimate obtained in the first step. In both of these steps, the underlying step is an optimization problem over the set of fixed-rank matrices. The overall algorithm is presented in Table I.

A. Finding Sparsity Pattern

In the first step, we reformulate problem (9) as the *regularized* problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \sum_{i=1}^K (X_{ii} - 1)^2 + \rho \sum_{ij} (X_{ij}^2 + \epsilon^2)^{1/2} \\ \text{subject to} \quad & \text{rank}(\mathbf{X}) = r, \end{aligned} \quad (10)$$

where $\rho \geq 0$ is the regularization parameter and ϵ is the parameter that approximates $|X_{ij}|$ with the smooth term $(X_{ij}^2 + \epsilon^2)^{1/2}$ that makes the objective function *differentiable*. A very small ϵ leads to ill-conditioning of the objective function in (10). Similarly, a larger ρ induces more sparsity in \mathbf{X} . Since we intend to obtain the sparsity pattern of the optimal \mathbf{X} , we set ϵ to a high value, e.g., 0.01, to make the problem (10) well conditioned.

If $\mathbf{X}_{\text{opt}} = [\mathbf{X}_{\text{opt}}]_{ij}$ is the solution of (10), then the sparsity pattern matrix $\mathbf{P} = [\mathbf{P}_{ij}]$ is of size $K \times K$ such that $P_{ij} = 1$ if $X_{\text{opt}ij} > \epsilon$ and $P_{ij} = 0$ otherwise.

TABLE I
RIEMANNIAN OPTIMIZATION ALGORITHM FOR \mathcal{P} .

- Finding sparsity partition: we solve the *regularized* formulation (10) to identify a good *sparsity* pattern \mathbf{P} , which is a binary matrix of size $K \times K$ with 1s at non-zero positions and 0s at zero positions.
- Refining: once the sparsity pattern \mathbf{P} is determined, we solve the matrix completion problem (11) with rank constraint to refine the estimate obtained from the regularized formulation solution.
- Both (10) and (11) are solved with a Riemannian trust-region algorithm on the set of fixed-rank matrices.

B. Refining the Estimate

Once the sparsity pattern \mathbf{P} is determined by solving (10), the *refining step* translates into solving a rank-constrained *matrix completion* problem. To see this, note that we know the positions of zeros in the solution matrix (from \mathbf{P}) and that the diagonal entries are all 1s. Consequently, computing the entries at other positions is *equivalent* to the problem

$$\begin{aligned} \underset{\mathbf{X} \in \mathbb{R}^{n \times n}}{\text{minimize}} \quad & \frac{1}{2} \sum_{i=1}^K (X_{ii} - 1)^2 + \frac{1}{2} \|(\mathbf{P} * \mathbf{X}) - \mathbf{X}\|_F^2 \quad (11) \\ \text{subject to} \quad & \text{rank}(\mathbf{X}) = r, \end{aligned}$$

where $\|\cdot\|_F$ is the *Frobenius* norm of a matrix and $\mathbf{P} * \mathbf{X}$ is the element-wise multiplication of the matrices \mathbf{P} and \mathbf{X} . Additionally, the algorithm for (11) is initialized from \mathbf{X}_{opt} , which is the solution of (10).

C. Fixed-Rank Riemannian Manifold Optimization

The optimization problems (10) and (11) are regularized *least-square* optimization problems over the set of fixed-rank matrices. A rank- r matrix $\mathbf{X} \in \mathbb{R}^{K \times K}$ is factorized as $\mathbf{X} = \mathbf{U}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{K \times r}$ and $\mathbf{V} \in \mathbb{R}^{K \times r}$ are full column-rank matrices. Such a factorization, however, is not unique as \mathbf{X} remains unchanged under the transformation of the factors

$$(\mathbf{U}, \mathbf{V}) \mapsto (\mathbf{U}\mathbf{M}^{-1}, \mathbf{V}\mathbf{M}^T), \quad (12)$$

for all non-singular matrices $\mathbf{M} \in \text{GL}(r)$, the set of $r \times r$ non-singular matrices. Equivalently, $\mathbf{X} = \mathbf{U}\mathbf{V}^T = \mathbf{U}\mathbf{M}^{-1}(\mathbf{V}\mathbf{M}^T)^T$ for all non-singular matrices \mathbf{M} . As a result, the local minima of an objective function parameterized with \mathbf{U} and \mathbf{V} are not isolated on $\mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r}$.

The classical remedy to remove this indeterminacy requires further (triangular-like) structure in the factors \mathbf{U} and \mathbf{V} . For example, LU decomposition is a way forward. In contrast, we encode the invariance map (12) in an abstract search space by optimizing directly over a set of equivalence classes

$$[(\mathbf{U}, \mathbf{V})] := \{(\mathbf{U}\mathbf{M}^{-1}, \mathbf{V}\mathbf{M}^T) : \mathbf{M} \in \text{GL}(r)\}. \quad (13)$$

The set of equivalence classes is termed as the *quotient space* and is denoted by

$$\mathcal{M}_r := \mathcal{M}/\text{GL}(r), \quad (14)$$

where the total space \mathcal{M} is the product space $\mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r}$.

Consequently, if an element $x \in \mathcal{M}$ has the matrix characterization (\mathbf{U}, \mathbf{V}) , then (10) and (11) are of the form

$$\underset{[x] \in \mathcal{M}_r}{\text{minimize}} \quad f([x]), \quad (15)$$

where $[x] = [(\mathbf{U}, \mathbf{V})]$ is defined in (13) and $f : \mathcal{M} \rightarrow \mathbb{R} : x \mapsto f(x)$ is a *smooth* function on \mathcal{M} , but now induced (with slight abuse of notation) on the quotient space \mathcal{M}_r (14).

The quotient space \mathcal{M}_r has the structure of a smooth *Riemannian* quotient manifold of \mathcal{M} by $\text{GL}(r)$ [13]. The Riemannian structure conceptually transforms a rank-constrained optimization problem into an *unconstrained* optimization problem over the non-linear manifold \mathcal{M}_r . Additionally, it allows to compute objects like gradient (of an objective function) and develop a Riemannian trust-region algorithm on \mathcal{M}_r that uses second-order information for faster convergence [12].

V. OPTIMIZATION ON QUOTIENT MANIFOLD

Consider an equivalence relation \sim in the *total* (computational) space \mathcal{M} . The quotient manifold \mathcal{M}/\sim generated by this equivalence property consists of elements that are *equivalence classes* of the form $[x] = \{y \in \mathcal{M} : y \sim x\}$. Equivalently, if $[x]$ is an element in \mathcal{M}/\sim , then its matrix representation in \mathcal{M} is x . Figure 2 shows a schematic viewpoint of optimization on a quotient manifold. Particularly, we need the notion of “linearization” of the search space, “search” direction and a way “move” on a manifold. Below we show the concrete development of these objects that allow to do develop a second-order trust-regions algorithm on manifolds.

Since the manifold \mathcal{M}/\sim is an abstract space, the elements of its tangent space $T_{[x]}(\mathcal{M}/\sim)$ at $[x]$ also call for a matrix representation in the tangent space $T_x\mathcal{M}$ that respects the equivalence relation \sim . Equivalently, the matrix representation of $T_{[x]}(\mathcal{M}/\sim)$ should be restricted to the directions in the tangent space $T_x\mathcal{M}$ on the total space \mathcal{M} at x that do not induce a displacement along the equivalence class $[x]$. This is realized by decomposing $T_x\mathcal{M}$ into complementary subspaces, the *vertical* and *horizontal* subspaces such that $\mathcal{V}_x \oplus \mathcal{H}_x = T_x\mathcal{M}$. The vertical space \mathcal{V}_x is the tangent space of the equivalence class $[x]$. On the other hand, the horizontal space \mathcal{H}_x , which is any complementary subspace to \mathcal{V}_x in $T_x\mathcal{M}$, provides a valid matrix representation of the abstract tangent space $T_{[x]}(\mathcal{M}/\sim)$ [12, Section 3.5.8]. An abstract tangent vector $\xi_{[x]} \in T_{[x]}(\mathcal{M}/\sim)$ at $[x]$ has a unique element in the horizontal space $\xi_x \in \mathcal{H}_x$ that is called its *horizontal lift*. Our specific choice of the horizontal space is the subspace of $T_x\mathcal{M}$ that is the *orthogonal complement* of \mathcal{V}_x in the sense of a Riemannian metric (an inner product).

A particular Riemannian metric on the total space \mathcal{M} that takes into account the symmetry (12) imposed by the factorization model, and that is well suited to a least-squares objective [15], is

$$g_x(\xi_x, \eta_x) = \text{Tr}((\mathbf{V}^T \mathbf{V}) \xi_{\mathbf{U}}^T \eta_{\mathbf{U}}) + \text{Tr}((\mathbf{U}^T \mathbf{U}) \xi_{\mathbf{V}}^T \eta_{\mathbf{V}}), \quad (16)$$

where $x = (\mathbf{U}, \mathbf{V})$ and $\xi_x, \eta_x \in T_x\mathcal{M}$. It should be noted that the tangent space $T_x\mathcal{M}$ has the matrix characterization $\mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r}$. Consequently, η_x (and similarly ξ_x) has the

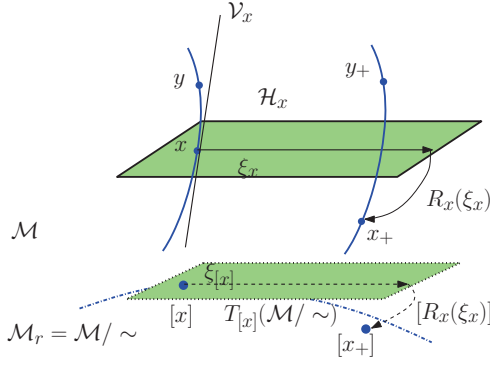


Fig. 2. Optimization on a quotient manifold. The dotted lines represent abstract objects and the solid lines are their matrix representations. The points x and y in the total (computational) space \mathcal{M} belong to the same equivalence class (shown in solid blue color) and they represent a single point $[x] := \{y \in \mathcal{M} : y \sim x\}$ in the quotient space \mathcal{M}/\sim . An algorithm by necessity is implemented in the computation space, but conceptually, the search is on the quotient manifold. Given a search direction ξ_x at x , the updated point on \mathcal{M} is given by the retraction mapping R_x .

matrix representation $(\eta_U, \eta_V) \in \mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r}$. Motivation for the metric (16) comes from the fact that it is induced from a block approximation of the Hessian of a least-squares objective function. Similar idea has also been exploited in [7].

Once the metric (16) is defined on \mathcal{M} , the development of the geometric objects required for second-order optimization follow [15]. The matrix characterizations of the tangent space $T_x\mathcal{M}$, vertical space \mathcal{V}_x , and horizontal space \mathcal{H}_x are straightforward with the expressions:

$$\begin{aligned} T_x\mathcal{M} &= \mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r} \\ \mathcal{V}_x &= \{(-U\Lambda, V\Lambda^T) : \Lambda \in \mathbb{R}^{r \times r}\} \\ \mathcal{H}_x &= \{(\zeta_U, \zeta_V) : U^T\zeta_U V^T V = U^T U \zeta_V^T V, \\ &\quad \zeta_U, \zeta_V \in \mathbb{R}^{K \times r}\}. \end{aligned} \quad (17)$$

Apart from the characterization of the horizontal space, we need a linear mapping $\Pi_x : T_x\mathcal{M} \mapsto \mathcal{H}_x$ that projects vectors from the tangent space onto the horizontal space. Projecting an element $\eta_x \in T_x\mathcal{M}$ onto the horizontal space is accomplished with the operator

$$\Pi_x(\eta_x) = (\eta_U + U\Lambda, \eta_V - V\Lambda^T), \quad (18)$$

where $\Lambda \in \mathbb{R}^{r \times r}$ is uniquely obtained by ensuring that $\Pi_x(\eta_x)$ belongs to the horizontal space characterized in (17). Finally, the expression of Λ is

$$\begin{aligned} U^T(\eta_U + U\Lambda)V^T V &= U^T U(\eta_V - V\Lambda^T)^T V \\ \Rightarrow \Lambda &= 0.5[\eta_V^T V(V^T V)^{-1} - (U^T U)^{-1}U^T \eta_U]. \end{aligned}$$

A. Gradient and Hessian Computation

The choice of the metric (16) and of the horizontal space (as the orthogonal complement of \mathcal{V}_x) turns the quotient manifold \mathcal{M}/\sim into a *Riemannian submersion* of (\mathcal{M}, g) [12, Section 3.6.2]. As shown in [12], this special construction allows for a convenient matrix representation of the gradient [12, Section 3.6.2] and the Hessian [12, Proposition 5.3.3] on the quotient manifold \mathcal{M}/\sim .

The Riemannian gradient $\text{grad}_{[x]}f$ of f on \mathcal{M}/\sim is uniquely represented by its horizontal lift in \mathcal{M} which has

the matrix representation

$$\begin{aligned} \text{horizontal lift of } \text{grad}_{[x]}f \\ = \text{grad}_x f = \left(\frac{\partial f}{\partial U} (V^T V)^{-1}, \frac{\partial f}{\partial V} (U^T U)^{-1} \right), \end{aligned} \quad (19)$$

where $\text{grad}_x f$ is the gradient of f in \mathcal{M} and $\partial f / \partial U$ and $\partial f / \partial V$ are the *partial derivatives* of f with respect to U and V , respectively.

In addition to the Riemannian gradient computation (19), we also require the directional derivative of the gradient along a search direction. This is captured by a *connection* $\nabla_{\xi_x} \eta_x$, which is the *covariant derivative* of vector field η_x with respect to the vector field ξ_x . The Riemannian connection $\nabla_{\xi_x} \eta_{[x]}$ on the quotient manifold \mathcal{M}/\sim is uniquely represented in terms of the Riemannian connection $\nabla_{\xi_x} \eta_x$ in the total space \mathcal{M} [12, Proposition 5.3.3] which is

$$\text{horizontal lift of } \nabla_{\xi_x} \eta_{[x]} = \Pi_x(\nabla_{\xi_x} \eta_x), \quad (20)$$

where $\xi_{[x]}$ and $\eta_{[x]}$ are vector fields in \mathcal{M}/\sim and ξ_x and η_x are their horizontal lifts in \mathcal{M} . Here $\Pi_x(\cdot)$ is the projection operator defined in (18). It now remains to find out the Riemannian connection in the total space \mathcal{M} . We find the matrix expression by invoking the *Koszul* formula [12, Theorem 5.3.1]. After a routine calculation, the final expression is [15]

$\nabla_{\xi_x} \eta_x = D\eta_x[\xi_x] + (A_U, A_V)$, where

$$\begin{aligned} A_U &= \eta_U \text{Sym}(\xi_V^T V)(V^T V)^{-1} + \xi_U \text{Sym}(\eta_V^T V)(V^T V)^{-1} \\ &\quad - U \text{Sym}(\eta_V^T \xi_V)(V^T V)^{-1} \\ A_V &= \eta_V \text{Sym}(\xi_U^T U)(U^T U)^{-1} + \xi_V \text{Sym}(\eta_U^T U)(U^T U)^{-1} \\ &\quad - V \text{Sym}(\eta_U^T \xi_U)(U^T U)^{-1} \end{aligned} \quad (21)$$

and $D\xi[\eta]$ is the Euclidean directional derivative $D\xi[\eta] := \lim_{t \rightarrow 0} (\xi_{x+t\eta_x} - \xi_x)/t$. $\text{Sym}(\cdot)$ extracts the symmetric part of a square matrix, i.e., $\text{Sym}(Z) = (Z + Z^T)/2$.

The directional derivative of the Riemannian gradient in the direction $\xi_{[x]}$ is given by the *Riemannian Hessian operator* $\text{Hess}_{[x]}f[\xi_{[x]}]$ which is now directly defined in terms of the Riemannian connection ∇ . Based on (20) and (21), the horizontal lift of the Riemannian Hessian in \mathcal{M}/\sim has the matrix expression:

$$\text{horizontal lift of } \text{Hess}_{[x]}f[\xi_{[x]}] = \Pi_x(\nabla_{\xi_x} \text{grad}_x f), \quad (22)$$

where $\xi_{[x]} \in T_{[x]}(\mathcal{M}/\sim)$ and its horizontal lift $\xi_x \in \mathcal{H}_x$. $\Pi_x(\cdot)$ is the projection operator defined in (18).

B. Retraction

An iterative optimization algorithm involves computing a search direction (e.g., negative gradient) and then “moving in that direction”. The default option on a Riemannian manifold is to move along geodesics, leading to the definition of the *exponential map*. Because the calculation of the exponential map can be computationally demanding, it is customary in the context of manifold optimization to relax the constraint of moving along geodesics. To this end, we define *retraction* $R_x : \mathcal{H}_x \rightarrow \mathcal{M} : \xi_x \mapsto R_x(\xi_x)$ [12, Definition 4.1.1]. A

natural update on the manifold \mathcal{M} is, therefore, based on the update formula $x_+ = R_x(\xi_x)$, i.e., defined as

$$\begin{aligned} R_U(\xi_U) &= U + \xi_U \\ R_V(\xi_V) &= V + \xi_V, \end{aligned} \quad (23)$$

where $\xi_x = (\xi_U, \xi_V) \in \mathcal{H}_x$ is a search direction and $x_+ \in \mathcal{M}$. It translates into the update $[x_+] = [R_x(\xi_x)]$ on \mathcal{M}/\sim .

C. Riemannian Trust-Region Algorithm

Analogous to trust-region algorithms in the Euclidean space [16, Chapter 4], trust-region algorithms on a Riemannian quotient manifold with guaranteed superlinear rate convergence and global convergence have been proposed in [12, Chapter 7]. At each iteration we solve the *trust-region sub-problem* on the quotient manifold \mathcal{M}/\sim . The trust-region sub-problem is formulated as the minimization of the *locally-quadratic* model of the objective function. The concrete matrix characterizations of Riemannian gradient (19), Riemannian Hessian (22), projection operator (18), and retraction (23) allow to use an *off-the-shelf* trust-region implementation on manifolds, e.g., in Manopt [14].

VI. SIMULATION RESULTS

In this section, we compare the proposed Riemannian optimization algorithm in Table I with the alternating minimization algorithm based on (8) for the sparse and low-rank optimization problem \mathcal{P} . For the alternating minimization algorithm, we need to solve a sequence of subproblems with non-smooth ℓ_1 -norm objective and an affine constraint (i.e., linear programming problem) for which we use CVX [11]. The maximum number of iterations of the proposed alternating minimization algorithm is set to be 50. For the proposed Riemannian algorithm, we set ϵ to a high value of 0.01. A good choice of ρ is 0.001 and is obtained by cross-validation. The Riemannian algorithm in Table I is implemented in Manopt [14]. The maximum number of trust-region iterations is set to 100. The Matlab codes are available at <https://bamdevmishra.com/codes/indexcoding>.

Consider a sparse and low-rank optimization problem \mathcal{P} with $K = 16$. (We consider a smaller size instance as CVX is too computationally expensive to run larger ones.) The achievable normalized data rate equals $1/\text{rank}(\mathbf{X})$, and the amount of normalized side information equals $(\|\mathbf{X}\|_0 - K)$, which measures the cache size. Therefore, the sparsity and low-rankness tradeoff in Figure 3 reveals the tradeoff between the amount of side information and the achievable data rate in the corresponding index coding problem. Furthermore, Figure 3 demonstrates that, by encoding the second-order information in the algorithm design, the trust-region Riemannian algorithm can achieve sparser solutions than the alternating minimization algorithm.

VII. CONCLUSION

In this paper, we proposed a new sparse and low-rank optimization modeling framework to characterize the tradeoff between the amount of the side information and the achievable data rate by revealing the sparsity and low-rankness tradeoff in

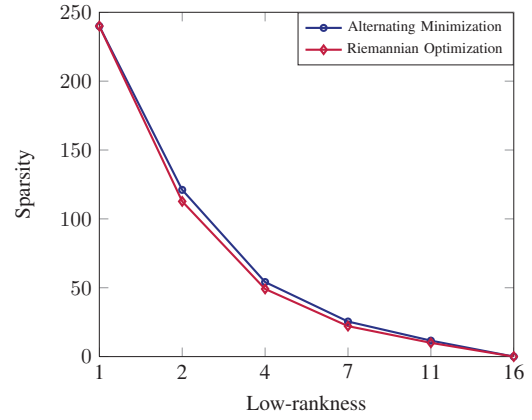


Fig. 3. Sparsity and low-rankness tradeoff in matrix \mathbf{X} , where sparsity is given by $(\|\mathbf{X}\|_0 - K)$ and the low-rankness is given by $\text{rank}(\mathbf{X})$.

the modeling matrix. A trust-region Riemannian optimization algorithm was proposed to improve the performance by encoding the second-order information, as well as the quotient manifold geometry of the fixed-rank matrices in the search space. This is achieved by relaxing the ℓ_0 -norm as a smooth ℓ_1 -norm surrogate and regularizing the affine constraint with least-squares objective. Simulation results revealed the fundamental tradeoff between the amount of side information and the achievable data rate in index coding problem. Our framework is useful for important system design problems, e.g., cache size allocation. A promising and interesting future research direction is theoretically characterizing the fundamental tradeoffs between storage size and the achievable data rate, i.e., the sparsity and low-rankness tradeoff in the proposed modeling matrix.

REFERENCES

- [1] S. Bi, R. Zhang, Z. Ding, and S. Cui, "Wireless communications in the era of big data," *IEEE Commun. Mag.*, vol. 53, pp. 190–199, Oct. 2015.
- [2] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled tactile internet," *IEEE J. Sel. Areas Commun.*, vol. 34, pp. 460–473, Mar. 2016.
- [3] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, pp. 2856–2867, May 2014.
- [4] H. Maleki, V. Cadambe, and S. Jafar, "Index coding—An interference alignment perspective," *IEEE Trans. Inf. Theory*, vol. 60, pp. 5402–5432, Sep. 2014.
- [5] Y. Shi, J. Zhang, K. Letaief, B. Bai, and W. Chen, "Large-scale convex optimization for ultra-dense Cloud-RAN," *IEEE Wireless Commun. Mag.*, vol. 22, pp. 84–91, Jun. 2015.
- [6] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, pp. 849–869, Feb. 2016.
- [7] Y. Shi, J. Zhang, and K. B. Letaief, "Low-rank matrix completion for topological interference management by Riemannian pursuit," *IEEE Trans. Wireless Commun.*, vol. PP, no. 99, 2016.
- [8] M. Effros, S. E. Rouayheb, and M. Langberg, "An equivalence between network coding and index coding," *IEEE Trans. Inf. Theory*, vol. 61, pp. 2478–2487, May 2015.
- [9] S. Oymak, A. Jalali, M. Fazel, Y. Eldar, and B. Hassibi, "Simultaneously structured models with application to sparse and low-rank matrices," *IEEE Trans. Inf. Theory*, vol. 61, pp. 2886–2908, May 2015.
- [10] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Program. Comp.*, vol. 4, no. 4, pp. 333–361, 2012.
- [11] CVX Research, Inc., "CVX: Matlab software for disciplined convex programming, version 2.0 (beta)," 2013.

- [12] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
- [13] B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre, “Fixed-rank matrix factorizations and Riemannian low-rank optimization,” *Comput. Statist.*, vol. 29, no. 3–4, pp. 591–621, 2014.
- [14] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, “Manopt, a Matlab toolbox for optimization on manifolds,” *J. Mach. Learn. Res.*, vol. 15, pp. 1455–1459, 2014.
- [15] B. Mishra, K. Adithya Apuroop, and R. Sepulchre, “A Riemannian geometry for low-rank matrix completion,” tech. rep., arXiv preprint arXiv:1211.1550, 2012.
- [16] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.